

REMARKS

At the outset, Applicants wish to thank Exr. Nguyen for her time during the recent telephone interview of April 29, 2008.

As the Examiner will recall, during the previous interview of October 5, 2007, attended by both Exr. Nguyen and SPE Vaughn, agreement was reached that the claims, as amended in the Amendment filed October 12, 2007, would overcome the applied §103(a) rejection based upon the then 5-way combination of references: 1) McLaughlin et al.; 2) Nazeem et al.; 3) Bowman-Amuah; 4) Mason; and 5) Bender et al.

Following submission of the Amendment filed October 12, 2007, the Office Action mailed January 9, 2008, issued a new §103(a) rejection based, this time, upon a 6-way combination of references: 1) McLaughlin et al.; 2) Nazeem et al.; 3) Bowman-Amuah; 4) Mason; 5) Hodges et al. and 6) Barker et al. Therefore, the Bender et al. reference in the previous rejection was replaced with Hodges et al. and Barker et al.

It is respectfully submitted that the Office is doing no more, at this point, than formulating §103(a) rejections by piling on references whose sole qualification for inclusion in the applied combination is that they contain some of the same words as are contained in the claims, presumably by a process that can only be described as key word searching, without regard to any valid technical or legal reasoning. During the telephone interview, Exr. Nguyen pointed to the recent U.S. Supreme Court's ruling in KSR International v. Teleflex as authority for such a rejection. However, it is respectfully submitted that KSR does not relieve the Office of its obligation to substantively examine the pending claims, and does not allow the Office to piece together a claimed embodiment from the mere presence of certain words or phrases in

references, when such words or phrases, are wholly unrelated to the art to which the claims pertain and would not provide the person of ordinary skill in the art any guidance or suggestion to achieve the claimed embodiments.

Claim 1 recites:

1. (Previously Presented) A method of servicing a request for a document over a computer network, comprising the steps of:

receiving a request for a document, the received document including a script that defines plurality of blocks, each block including a reference to a data source and code that is adapted to access the data source and to format the data accessed from the data source;

retrieving only some of the plurality of blocks defined in the script of the requested document from a memory, the memory storing the at least one of the plurality of blocks defined in the script of the requested document;

dynamically generating remaining ones of the plurality of blocks defined in the script of the requested document that were not retrieved from the memory and storing a copy of each dynamically generated block in the memory;

assembling the requested document from both the retrieved and dynamically generated blocks, and

sending the assembled document over the computer network to an originator of the request.

Now, kindly consider the two newly-added references to Hodges et al. and Barker et al.

Starting with Hodges et al., the outstanding Office Action stated that this reference teaches “generating remaining blocks” and points to Col. 9, line 49 to col. 10, line 8 as evidence thereof. However, Hodges et al. is wholly unconcerned with assembling a requested document from blocks retrieved from a memory and dynamically generated blocks. Instead, Hodges et al. teaches to reconstruct data in a Redundant Array of Inexpensive Disks (RAID) in the presence of multiple initiator commands. To do so, a regenerate command is issued to regenerate data from a drive that may not be available, using an XOR generator to XOR all of the remaining blocks in a parity stripe, as disclosed at Col. 10, lines 5-8:

5 to be reconstructed. The initiator uses its XOR generator to XOR all of the remaining blocks in the parity stripe, B0, B2, and P0 to regenerate the data block B1. The reconstructed block B1 is sent to the requesting host.

Although wholly unconcerned with assembling a requested document from blocks retrieved from a memory and dynamically generated blocks as claimed herein, the above-excerpted passage does include the terms “remaining blocks” and “regenerate” (which is close to the claimed term “generate”). It is respectfully submitted to the Office that the mere presence of the words “remaining blocks” and “regenerate” alone, without any technical or legal reasoning, is an insufficient basis on which to formulate a §103(a) rejection. Indeed, that some host uses an XOR generator to exclusive-OR all of the remaining blocks in a parity stripe of a RAID does not, whether considered singly or in combination with references 1-4, teach or suggest the claimed embodiment to a person of ordinary skill in the art. Specifically, a person of ordinary skill in the art, even in full possession of references 1-4 and 6, would not be motivated to carry out a step of

dynamically generating remaining ones of the plurality of blocks defined in the script of the requested document that were not retrieved from the memory and storing a copy of each dynamically generated block in the memory;

simply because Hodges et al.’s patent contained the words “remaining blocks” and “regenerate.” In a blatant *non sequitur*, the Office stated that a person of ordinary skill would somehow be motivated to incorporate Hodges et al.’s teachings “for the purpose of providing large storage capabilities and high reliability at a low cost.” However, this reason makes no sense at all in the context of the claimed embodiments, as the claimed embodiments do not provide “large storage capabilities and high reliability at a low cost,” nor would such provide any motivation to a person of skill to develop the claimed embodiments. The claimed embodiments are drawn to servicing requests for documents over a computer network. Large storage capabilities and high

reliability at low cost are characteristics of RAID arrays, which is the subject matter of Hodges et al.

It is respectfully submitted, therefore, that Hodges et al. is wholly unrelated to the claimed subject matter and was included as a reference in the applied §103(a) solely by virtue of its inclusion, in its specification, of the words “remaining blocks” and “regenerate,” which is an insufficient basis for a §103(a) rejection, even in our post-KSR world.

The next reference to Barker et al. is similarly technologically unrelated to the claimed embodiments, and has also been included in the applied §103(a) rejection solely by virtue of its inclusion of words that also happen to appear in the pending claims.

The Office stated that Barker et al. “discloses assembling the requested document from both the retrieved and dynamically generated blocks,” and referred the reader to Barker et al.’s paragraph [0037], which is reproduced below:

[0037] The client executes the Client Interface and propriety applications via Web pages. Microsoft Internet Explorer and netscape browsers are supported as are the web-enabling devices for PCs and X-terminals. Through a Web-based graphical client interface, clients’ commands generate HTTP requests to the element management system server. The server gathers information, dynamically generates a Web page, and sends the results/output to the web browser for display.

At the outset, Barker et al. do not, in paragraph [0037], teach or even remotely suggest **“assembling the requested document from both the retrieved and dynamically generated blocks.”** There is no teaching of assembling any document from both retrieved and dynamically generated blocks, and saying so in an Office Action does not, without more, make it so.

What is taught in this paragraph is that a Web server “dynamically generates a Web page.” The phrase “dynamically generates a Web page” sounds similar to the claimed

dynamically generating remaining ones of the plurality of blocks defined in the script of the requested document that were not retrieved from the memory and storing a copy of each dynamically generated block in the memory;

However, Hodges et al. simply teaches that a server dynamically generates a Web page, which is ... what Web servers do. In response to HTTP requests issued from clients, Web servers dynamically generate Web pages, as has been known for many years now. There is no teaching or suggestion, in this reference, of the claimed subject matter or of the subject matter acknowledged to be missing in references 1-4.

Therefore, to piece together the claimed language:

dynamically generating remaining ones of the plurality of blocks ...

the Office used two separate references; namely, Barker et al. for “**dynamically generating**” and Hodges et al. for “**remaining blocks**” to come up with “**dynamically generating remaining block,**” which sounds very similar to the claim language. To support the use of Barker et al., the Office relied upon on old standby: “for the purpose of enhancing efficiency and convenience,” which - again - makes no sense in the context of the claimed embodiment and would provide no motivation whatsoever for the skilled artisan seeking to devise a method for servicing a request for a document over a computer network, as claimed herein.

Even KSR does not provide the Office with the unbridled authority to combined unrelated and incompatible references because they happen to include some of the same words that are used in the claims. KSR did allow an Examiner to look to non-analogous art in certain circumstances, but neither the Supreme Court nor the KSR decision KSR intended to eviscerate

§103(a) by wholly removing therefrom the requirement that the invention as a whole must have been obvious to a person of ordinary skill in the art at the time the invention was made. In this case, KSR does not allow the Office to ignore the POSITA test in favor of a mechanistic reconstruction of the claim language based upon key word searches yielding no more than isolated and unrelated snatches of language that are then stitched together to somehow resemble the claim language.

For the sake of completeness, the arguments below deal with references 1 through 4, and will demonstrate that adding Hodges et al. and/or Barker et al. to the mix would not teach or suggest the claimed embodiment to a person of ordinary skill in the art.

As the Examiner will recall, claim 1 requires that only some of the blocks be retrieved from the memory and that remaining ones of the plurality of blocks be dynamically generated. The claimed embodiment, therefore, requires two actions: 1) retrieving only some of the blocks from the memory; and 2) dynamically generating remaining ones of the plurality of blocks that were not retrieved from the memory. The applied combination does not teach or suggest such a method.

The applied combination, however, does not teach or suggest that only some of the blocks be retrieved from a memory and that all remaining blocks defined in the script be dynamically generated, as claimed herein. Indeed, the primary reference to McLaughlin et al. teaches an all or nothing approach to serving a client application with process data:

When a client application requires process data, a data request is sent to supervisory controller 120 and is received in cache manager 220. In one embodiment of the present invention, cache manager 220 first searches dynamic cache 220 for the requested process data. If the process data is found (a cache hit), cache manager 220 transfers the process data to the requesting client application and the transaction is ended. If the data is not found (a cache miss), cache manager 220 then requests the process data from the appropriate one of process nodes 204–206. When the supervisory controller 120 receives the process data from the process node, cache manager 215 writes the process data into dynamic cache 215 and transfers the process data to the client application that originally requested it.

as shown in Col. 8, lines 6-19. Either process data is present in the cache 220 or it is fetched from one of the process nodes 204-206. No provisions are made within McLaughlin et al. for retrieving only some of the blocks from the memory and dynamically generating remaining ones of the plurality of blocks that were not retrieved from the memory. Alternatively, McLaughlin et al. teach that process data may be obtained in a peer-to-peer fashion, as detailed in Col. 6, lines 22-25 or via a publish/subscribe model, as detailed in Col. 3, lines 55-57.

Similarly, all of the supervisory data is dynamically generated, as positively stated in Col. 5, lines 33-36:

optimize the facility as a whole. In a preferred embodiment, the supervisory data is dynamically generated and is based at least upon a given facility's efficiency, production or economic cost, and most preferably all three.

which means that none of the supervisory data is retrieved from a cache memory. Therefore, McLaughlin et al. do not teach or suggest retrieving only some of the blocks from the memory and dynamically generating remaining ones of the plurality of blocks that were not retrieved from the memory, as required by the amended claims. Similarly, the McLaughlin et al. reference cannot teach the claimed assembling step, as such requires that the requested document be assembled from both the retrieved and generated blocks.

The outstanding Office Action again states that Nazeem et al. teach a document that includes a plurality of blocks. However, Nazem et al. do not remedy the fundamental shortcomings of the primary reference and do not teach – alone or in combination with McLaughlin et al. - to retrieve only some of the blocks from the memory and to dynamically generate remaining ones of the plurality of blocks, as claimed herein. Indeed, Nazem et al. teach the use of templates to hold the static data and to obtain all live data from the shared memory 212:

in further detail. User front page 218 is built according to a user template and live data. The user template specifies, for example which quotes are shown in the portfolio module, which cities are displayed in the weather module, etc. Each of the modules 504 can be customized by a user and moved about front page 218. The modules 504 are also reusable, in that any customized module which appears on multiple pages can be edited from any one of those pages and the edits will be reflected on each of the pages. Other custom pages for the user can be viewed by selecting one of the page buttons 502 appearing below the header. Other pages and utilities can be selected using the buttons 508 which are part of the header.

In addition to all of the live data shown in FIG. 5 being stored in the shared memory, summaries from each of the

as stated in line 66-67 above. Nazem et al. teach that the template data is stored (and necessarily retrieved from) the cached user templates database 214 and that all of the live data to fill these templates is stored in (and necessarily retrieved from) the shared memory 212. Therefore, even when considered collectively with the McLaughlin et al. reference, Nazem et al. do not teach or suggest to retrieve only some of the blocks from the memory and to dynamically generate remaining ones of the plurality of blocks that were not retrieved from the memory, as required by each of the independent claims herein. The assembling step is also wholly unsuggested by the combination.

It is respectfully submitted that adding the Bowman-Amuah reference to the mix does not provide any additional guidance to the person of ordinary skill in this art. The cited passages

simply do not teach anything relevant to blocks of a document including a reference to a data source and code that is configured to access and format data accessed from the data sources, as required by the claims. Indeed, Col. 52, lines 55-61 deals with ID and password pairs:

55 resources, as opposed to securing an applications detailed functions.

The security component prevents unauthorized users from accessing corporate data/resources by providing the users with access codes—password & ID—that allows the user to
60 login to the system or execute any (or a particular) application.

while Col. 47, lines 30-67 deals with wholly unrelated performance issues:

How important is performance? .40

In general, performance of data access and printing should be considered. Some typical benchmark tests include table scan, single-table report, joined table report, and mailing label generation times. (source is market research)

What is the budget? 35

Per developer costs as well as run time licensing fees, maintenance costs, support fees, and upgrade charges should be considered.

Do I have another component that satisfies this requirement? 40

Many databases and application development tools are shipped with built in or add-on report writing capability. However, stand-alone report writers: (1) are more powerful and flexible, especially when dealing with multiple data sources and a wide variety of formats; (2) can retrieve information from more data sources than the bundled report writers and can create reports from several data sources simultaneously; (3) excel in ease of use, both in designing and generating reports; (4) offer better tools and more predefined reports; and (5) have faster engines. (source is market research) 45
50

Does the product integrate with the existing or proposed architecture?

It is important to consider how well a product integrates with desktop tools (word processing, spreadsheet, graphics etc.) and application development programs. These items can be used to extend the capabilities of the reporting package. 55

What databases does the product support? 60

A product should support the most widely used PC file formats and Client/Server databases. It may be necessary to consider the type of support. For example, native database interfaces tend to have better performance than open standards such as ODBC. Another possible consideration is how well the product accesses multiple files or databases. (source is market research) 65

Next, the Office turns to Mason, Jr. for a supposed teaching of “retrieving only some of the plurality of blocks, and generating remaining blocks,” and points to Col. 8, lines 1-28 and Col. 9, lines 1-30 in support of its §103(a) rejection. However, Mason, Jr., whether considered alone or in combination with McLaughlin et al., Nazem et al. (5,983,227) and/or Bowman-Amuah does not teach or suggest the claimed subject matter.

Indeed, Mason, Jr. (similarly to newly-cited Hodges et al) teaches a RAID controller system, and not a method of servicing a request for a document over a computer network, as claimed. Mason, Jr. teaches, at Col. 8, lines 1-30 referred to by the Examiner, that in case of a read operation failure, the data is reconstructed using a corresponding parity block of the stripe in which the failure occurred. The back end cache must then determine whether the parity block has already been cached by referring to a back end cache block list. If it is, then a read I/O operation may be avoided. Any remaining blocks of the stripe are read and cached in a front end cache. Then all blocks of the stripe in which the failure occurred are XORed together with the parity block, whereupon valid data is passed to the front end cache. No teaching or suggestion are present in this passage of Mason, Jr., whether considered alone or in combination with the other references, that would lead a person of ordinary skill in the art to the claimed embodiments – that is, to retrieve only some of the blocks from the memory and to dynamically generate remaining ones of the plurality of blocks that were not retrieved from the memory, as claimed herein. Likewise, this combination cannot teach the claimed assembling step.

Next, the Office points to Col. 9, lines 1-30 of Mason, Jr. This passage teaches a write command in which a list of valid blocks is checked. The list (see Col. 8, lines 56-61) accompanies a write command received from a host, includes the new blocks to be written to the RAID array. Mason, Jr. teaches, in Col. 9, beginning at line 4, teaches that if some old blocks to be written are missing from the list (meaning that a cache miss has occurred), they are retrieved from the physical disk and stored into the front end cache. In essence, Mason, Jr. teaches that if blocks to be written are missing from the list, they are retrieved from the physical disks. At the outset, kindly note that this reference only teaches operations carried out within a RAID array of

hard disk drives, and does not teach servicing a request for a document over any computer network. Moreover, note that Mason, Jr. teaches that if blocks to be written are missing from the list, they are retrieved from the physical disks, which cannot be considered to teach or to suggest dynamically generating remaining ones of the plurality of blocks ... that are not stored in the memory. When blocks are missing from the list of blocks to be written, Mason, Jr. teaches simply to retrieve them from the disk. The claimed embodiment, on the other hand, requires dynamically generating remaining ones of the plurality of blocks, which is unsuggested by Mason, Jr., whether considered alone or in combination with the other constituent references of the applied combination. Moreover, it is respectfully submitted that the skilled artisan would not look to or find any guidance in a patent relating to a RAID controller when seeking to develop a method for servicing requests for documents over a computer network, as is the Mason, Jr. patent.

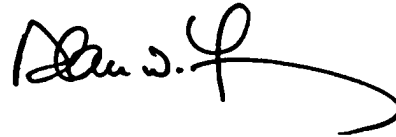
Kindly recall that the four-way McLaughlin et al.-Nazem et al.-Bowman-Amuah-Mason, Jr. combination was overcome by the Amendment filed November 4, 2006. The Office added Bender et al. to the applied combination of references, which was subsequently replaced by **Hodges et al.** and **Barker et al.** in the outstanding Office Action. Therefore, adding Hodges et al. and Barker et al. to the previously overcome 4-way combination, for at least the reasons advanced above, does not remedy the fundamental shortcomings of the primary 4-way combination of references.

Claims 19, 37 and 55 and their respective dependent claims are believed to be unsuggested by the applied 6-way combination for the same reasons.

As the applied combination of references does not teach or suggest the claimed embodiments, reconsideration and withdrawal of the outstanding §103(a) rejections are respectfully requested.

Applicants believe that this application is now in condition for allowance. If any unresolved issues remain, please contact the undersigned attorney of record at the telephone number indicated below and whatever is necessary to resolve such issues will be done at once.

Respectfully submitted,



Date: May 9, 2008

By: _____

Alan W. Young
Attorney for Applicants
Registration No. 37,970

YOUNG LAW FIRM, P.C.
4370 Alpine Rd., Ste. 106
Portola Valley, CA 94028
Tel.: (650) 851-7210
Fax: (650) 851-7232

C:\YLF\CLIENTS\ORCL\5680\5680 AMEND.6.doc